Chima Full Stack Engineer Case Study: AI Video Ad Generator

Project Overview

Time Limit: 24 Hours

Build a competitor to Creatify.ai focusing on the **URL to Video** feature - an AI-powered platform that transforms any product page URL into compelling video advertisements.

Core Concept

Users input a product URL → AI analyzes the page → Generates professional video ads automatically

Technical Requirements

Minimum Viable Product (MVP) Features

1. **URL Input & Scraping**
   - Accept product page URLs
   - Extract product images, descriptions, and key features
   - Focus on 1-2 major platforms (e.g., Shopify stores or Amazon)
2. **AI-Powered Content Generation**
   - Generate compelling ad copy using LLMs
   - Create at least one script variation
   - Extract key product benefits
3. **Video Generation**
   - Convert scraped content into video format
   - Include product images
   - Add text overlays with basic animations
   - Duration: 15-30 seconds
   - At least one aspect ratio (9:16 for vertical or 16:9 for horizontal)
4. **User Interface**

   - Simple, functional dashboard
   - URL input form
   - Video preview
   - Download option

Technical Stack Recommendations

Frontend:

- React/Next.js or Vue.js
- Any CSS framework (Tailwind recommended)
- Basic video player for preview

Backend:

- Node.js/Express or Python/FastAPI
- Simple file-based storage or SQLite (no need for complex DB)
- Local processing (no need for deployment)

AI/ML Services (Use Your Own Keys):

- OpenAI API or Anthropic Claude for text generation
- Basic text-to-speech (optional)
- FFmpeg or Remotion for video generation

24-Hour Implementation Plan

Hours 1-4: Foundation

- Set up project structure
- Implement basic URL scraping
- Create simple UI with URL input
- Test data extraction from product pages

Hours 5-10: AI Integration

- Integrate LLM API for script generation
- Process scraped product data
- Generate video script/storyboard
- Create basic content templates

Hours 11-18: Video Generation

- Implement video composition
- Add text overlays
- Include product images
- Generate MP4 output

Hours 19-24: Polish & Demo

- Refine UI
- Handle basic errors
- Create demo videos
- Record Loom walkthrough
- Push to GitHub

Evaluation Criteria

Functionality (50%)

- URL scraping works
- AI generates relevant ad copy
- Video is successfully created
- End-to-end pipeline functions

Code Quality (25%)

- Clean, readable code
- Basic error handling
- Clear project structure
- README with setup instructions

Innovation & Problem Solving (25%)

- Creative approach to challenges
- Quality of AI-generated content
- Video output quality
- UI/UX considerations

Deliverables

1. **GitHub Repository**
    - Complete source code
    - README with:
        - Setup instructions
        - Required API keys list
        - How to run locally
    - Sample `.env.example` file
2. **Loom Video (5-10 minutes)**
    - Live demonstration of the platform
    - Show complete user flow:

        - Input URL
        - View generation process
        - Download final video
    - Brief code walkthrough
    - Challenges faced and solutions
3. **Sample Output**

    - Include 2-3 generated videos in the repo
    - Show different product types if possible

Focused Scope for 24 Hours

Must Have

- URL → Video pipeline working
- One video template/style
- Basic UI
- Local functionality (no deployment needed)

Nice to Have (If Time Permits)

- Multiple aspect ratios
- Voice-over
- Multiple templates
- Advanced animations

Skip (Out of Scope)

- User authentication
- Database persistence
- Payment integration
- Complex video editing features
- Backend deployment

Tips for Success

1. **Use AI Coding Assistants**: Leverage Cursor, GitHub Copilot, or Claude to speed up development
2. **Start Simple**: Get a basic version working first, then enhance
3. **Pre-built Libraries**: Use existing libraries for heavy lifting (video generation, animations)
4. **Focus on Demo**: Make sure you have something impressive to show, even if not all features are complete

Quick Start Resources

Recommended Libraries

- **Web Scraping**: Puppeteer, Playwright (JavaScript) or BeautifulSoup (Python)
- **Video Generation**:
  - Remotion (React-based)
  - MoviePy (Python)
  - FFmpeg with Node.js wrapper
- **AI Integration**: OpenAI or Anthropic SDK

Sample Architecture

Unset

```
Frontend (React)
    ↓
Backend API
    ↓
Scraping Service → AI Service → Video Generator
    ↓                  ↓              ↓
Product Data     Ad Script      Final Video
```

Common 24-Hour Pitfalls to Avoid

1. Spending too much time on perfect scraping - focus on 1-2 sites
2. Over-complicating video generation - simple works
3. Getting stuck on deployment - local demo is sufficient
4. Perfecting UI - functional beats beautiful for this timeline

Submission Checklist

- [ ] GitHub repository is public/accessible
- [ ] README includes clear setup instructions
- [ ] `.env.example` shows required API keys
- [ ] Loom video demonstrates full functionality
- [ ] Code runs locally without deployment
- [ ] At least one sample video included

Remember: The goal is to demonstrate your ability to integrate multiple technologies and solve problems quickly. A working prototype that shows the core concept is better than an incomplete perfect solution. Good luck!